In the claims: (the following listing replaces all prior versions):

1. (Cancelled) ~~A method for use in a distributed system for processing a mobile program that has the ability to move from node to node in the distributed system comprising~~

~~including, in an a mobile program operating environment in each of the nodes, providing service facilities useful to the mobile program, and~~

~~in the operating environment, in each of the nodes, running a supervisor process that allows the mobile program indirect access to make use of the service facilities.~~

2. (Currently amended) ~~The method of claim 1 further comprising~~ A method for providing service facilities to a mobile program in a distributed system having nodes, comprising

in an operating environment in each of the nodes of the distributed system, providing service facilities useful to the mobile program,

in the service facilities , running a supervisor process that allows the mobile program indirect access to make use of the node's operating environment,

the supervisor process creating a bastion object ~~in an unrestricted environment~~ to protect the unrestricted environment and ~~passing it into~~ running the bastion object in a restricted environment within which the mobile program is running.

3. (Previously presented) The method of claim 2 in which the bastion object provides an interface for the mobile program to access the service facilities in a safe manner and which is substantially the same interface as the interface that the service facilities provide in the unrestricted environment.

4. (Previously presented) The method of claim 2 in which the bastion object performs type checking on all method calls made by a mobile program to a service facility.

5. (Previously presented) A method for use in a distributed system for processing a mobile program that executes in one node of the distributed system, may be interrupted at almost any point in its execution, and may be moved to another node of the distributed system for further execution, comprising

in the one node, capturing a current state of the mobile program execution,

delivering the captured state and program code of the mobile program to the other node, and

continuing execution at the other node from the point of interruption based on the captured state and the program code.

6. (Previously presented) The method of claim 5 further comprising

also delivering with the captured state and the program code a transported file system or other information created during execution of the mobile program.

7. (Previously presented) The method of claim 6 in which the information in the transported file system or other information is accessible without executing the mobile program.

8. (Previously presented) The method of claim 5 in which the step of capturing comprises using an encoding scheme of a language interpreter.

9. (currently amended) A method for enabling communication with a mobile program running in a distributed system, a mobile program service station, an extension, or another application, comprising

providing a mechanism which permits each of mobile program, the mobile program service station, the extension, and the other application to identify services that it ~~is~~ provides, and permits each of them to find services that it needs, and

enabling the mobile program to communicate with mobile program service stations via objects associated with the mechanism.

10. (Previously presented) The method of claim 9 in which each of the objects is provided by a supervisor process running in the distributed system and prevents uncontrolled access to a needed service.

11. (Previously presented) The method of claim 9 in which the mechanism includes a broker and a manager.

12. (Previously presented) The method of claim 9 in which the objects are data typed.

13. (Previously presented) A method for enabling negotiation between two unrelated mobile programs, mobile service stations, extensions, or other applications, in a distributed system, comprising

in an operating environment in a node of the distributed system, receiving information from one of the two mobile programs, mobile program service stations, extensions, or other applications, concerning a transaction offered to other mobile programs, mobile program service stations, extensions, or other applications,

in the operating environment in the node, receiving information from the second of the two mobile programs, mobile programs service stations, extensions, or other applications concerning a transaction in which the second of the mobile programs, mobile program service stations, extensions, and other applications wishes to engage,

notifying the second mobile program, mobile program service station, extension, or other application of the one mobile program, mobile program service station, extension, or other application, and

enabling the two mobile programs, mobile program service stations, extensions, or other applications to communicate concerning the transaction.

14. (Previously presented) The method of claim 13 in which the information is received from two mobile programs by a third mobile program.

15. (Previously presented) A method for enabling action by an operating environment in a distributed system with respect to a mobile program which is programmed in a language that is not fully supported by the operating environment, comprising

labeling a mobile program to identify operating environment features required for full support of the mobile program,

in an operating environment, examining the labeling of the mobile program to determine whether the operating environment supports all of the identified features, and

taking an action based on whether all the identified features are supported.

16. (Previously presented) The method of claim 15 wherein the action comprises sending the mobile program to another operating environment for processing.

17. (Previously presented) The method of claim 15 in which the action comprises retrieving nonprogram specific data from the mobile program.

18. (currently amended) A method for aiding communication with a mobile program executing in ~~operating environments provided at nodes of~~ a distributed system, comprising

maintaining a name space that uniquely identifies types of information to be interchanged as part of the communication, and

using a name within the name space to identify a type of information to be interchanged.

19. (Previously presented) The method of claim 18 in which the mobile program registers an interface which includes the name of a type of information that is to be interchanged.

20. (Cancelled)

21. (Cancelled)

22. (Cancelled)

23. (Cancelled)

24. (Previously presented) A method for controlling interaction between a mobile program and an application running in an operating environment provided at a node of a distributed system, comprising

defining a trusted portion of the operating environment which provides trusted services to the mobile program,

requiring portions of the application running in the operating environment to be registered as trusted, and

permitting indirect interaction via the operating environment between the mobile program and the application running in the operating environment only if the portions of the application required to be registered have been registered.

25. (Previously presented) A method for enabling a mobile program to carry out defined functions including otherwise unsafe functions, through the use of extensions comprising

coding safe extensions to an operating environment and to an interpretive language under which the mobile program runs, and

permitting the mobile program to carry out the defined functions by making use of the extensions.

26. (Previously presented) The method of claim 9 in which the mechanism comprises a connector mechanism, and the objects comprise connector objects.

27. (Currently amended) The method of claim ~~1~~ 2 in which the service facilities of the operating environment are used to access other resources not part of the operating environment.

28. (Currently amended) The method of claim ~~1~~ 2 in which the supervisor process prohibits unauthorized access to the service facilities of the operating environment.